

Software Engineering Methodology

Chapter 8.0 **Software Integration and Testing Stage**

Table of Contents

Chapter		Page
8.0	Software Integration and Testing Stage	8.0-1
8.1	Conduct Integration Testing	8.1-1
8.2	Conduct System Testing	8.2-1
8.3	Initiate Acceptance Process	8.3-1
8.4	Conduct Acceptance Test Team Training	8.4-1
8.5	Revise Project Plan	8.5-1
8.6	Conduct In-Stage Assessment	8.6-1
8.7	Conduct Software Integration and Testing Stage Exit	8.7-1

Stage: **8.0**
Software Integration and Testing Stage

Description: Software integration and testing activities focus on interfaces between and among components of the software product, such as functional correctness, system stability, overall system operability, system security, and system performance requirements (e.g., reliability, maintainability, and availability). Software integration and testing performed incrementally provides feedback on quality, errors, and design weaknesses early in the integration process.

In this stage, software components are integrated and tested to determine whether the software product meets predetermined functionality, performance, quality, interface, and security requirements. Once the software product is fully integrated, system testing is conducted to validate that the software product will operate in its intended environment, satisfies all user requirements, and is supported with complete and accurate operating documentation.

Input: The following items provide input to this stage.

- Project File
- Acceptance Test Plan (*draft*)
- Acquisition Plan
- Installation Plan (*draft*)
- Software modules
- Requirements Traceability Matrix (*expanded*)
- Project Test File
- Development baselines
- Transition Plan
- Operating Documentation (*draft*)
 - Users Manual
 - Programmers Reference Manual
- Training Plan (*draft*)
- Integration Test Plan
- System Test Plan
- Project Plan
- Software Quality Assurance Plan

High-Level Activities: The remainder of this chapter is divided into sections that describe specific high-level activities performed during this stage. These activities represent the minimum requirements for a large software engineering effort. *Notes* are provided, as applicable, to assist in customizing these lifecycle stage

**High-Level
Activities,
continued:**

requirements to accommodate the different sizes of software engineering efforts. The high-level activities are presented in the sections listed below.

- 8.1 Conduct Integration Testing
- 8.2 Conduct System Testing
- 8.3 Initiate Acceptance Process
- 8.4 Conduct Acceptance Test Team Training
- 8.5 Revise Project Plan
- 8.6 Conduct In-Stage Assessment
- 8.7 Conduct Software Integration and Testing Stage Exit

Output:

Several work products are produced during this stage. The work products listed below are the minimum requirements for a large software project. Deviations in the content and delivery of these work products are determined by the size and complexity of the project. Explanations of the work products are provided under the applicable activities described in the remainder of this chapter.

- Integration Test Reports
- System Test Report
- Operating Documents (*final*)
 - Users Manual
 - Programmers Reference Manual
- Training Plan (*final*)
- Installation Plan (*final*)
- Acceptance Test Plan (*final*)
- Preacceptance Checklist
- Security Checklist
- Error Reporting and Tracking System (*optional*)
- Project Plan (*revised*)

A matrix showing the work products associated with each high-level activity is provided in *Exhibit 8.0-1, Software Integration and Testing Stage Activities and Work Products by Project Size*. The matrix also shows which work products are deliverables and whether they are required or optional for small, medium, and large projects.

Review Process:

Structured walkthroughs are necessary during this stage to validate work products. The activities that are appropriate for structured walkthroughs are identified throughout the chapter. The time and resources needed to conduct the walkthroughs should be indicated in the project resources, schedule, and work breakdown structure.

Reference: *Appendix C, Conducting Structured Walkthroughs*, provides a procedure and sample forms that can be used for structured walkthroughs.

Bibliography: The following materials were used in the preparation of the Software Integration and Testing Stage chapter.

1. The Institute of Electrical and Electronics Engineers, Inc., *IEEE Standard for Developing Software Life Cycle Processes*, IEEE Std 1074-1991, New York, 1992.
2. U.S. Department of Commerce, National Institute of Standards and Technology, *Guide to Software Acceptance*, 500-180, Washington, D.C., 1990.
3. U.S. Department of Labor, Directorate of Information Resources Management, *Systems Engineering Concepts and Procedures Manual*, 1988.
4. U.S. Department of Labor, Directorate of Information Resources Management, *Systems Engineering Standards Manual*, 1988.

Exhibit 8.0-1. Software Integration and Testing Stage Activities and Work Products by Project Size

Work Activity		Project Size L M S			Work Product	Scheduled Deliverables L M S		
8.1	Conduct Integration Testing	R	R	R	Integration Test Materials Integration Test Report	R R	R R	R R
8.2	Conduct System Testing	R	R	R	System Test Materials System Test Report Operating Documents <i>(final)</i> Training Plan <i>(final)</i> Installation Plan <i>(final)</i>	R R R R R	R R R R R	R R R R A
8.3	Initiate Acceptance Process	R	R	R	Acceptance Test Plan <i>(final)</i> Preacceptance Checklist Security Checklist Error Reporting and Tracking System <i>(optional)</i>	R R A O	R R A O	R R A O
8.4	Conduct Acceptance Test Team Training	A	A	A				
8.5	Revise Project Plan	R	R	A	Project Plan <i>(revised)</i>	R	R	A
8.6	Conduct In-Stage Assessment	R	R	A	ISA Report Form ¹	N	N	N
8.7	Conduct Software Integration and Testing Stage Exit	R	R	A	Stage Exit Meeting Summary	N	N	N

Size: L = Large
M = Medium
S = Small

Minimum Requirements: R = Required
A = As Appropriate
N = Not Applicable

O = Optional work product
¹ = Completed by reviewer

Activity: **8.1**
Conduct Integration Testing

Responsibility: Project Team Programmers

Description: During software integration, the software components developed by the programming staff, off-the-shelf software purchased from vendors, and reusable code or modules obtained from other sources are assembled into one software product. Each assembly is tested in a systematic manner in accordance with the Integration Test Plan. An incremental approach to integration enables verification that as each new component is integrated, it continues to function as designed and both the component and the integrated product satisfy their assigned requirements.

Integration testing is a formal procedure that must be carefully planned and coordinated with the completion dates of the unit-tested modules. Integration testing begins with a software structure where called sub-elements are simulated by stubs. A stub is a simplified program or dummy module designed to provide the response (or one of the responses) that would be provided by the real sub-element. A stub allows testing of calling program control and interface correctness. Stubs are replaced by unit-tested modules or builds as integration testing proceeds. This process continues one element at a time until the entire system has been integrated and tested.

Integration testing may be performed using "bottom up" or "top down" techniques. Most integration test plans make use of both bottom-up and top-down techniques. Scheduling constraints and the need for parallel testing will affect the test approach.

The bottom-up approach incorporates one or more modules into a build; tests the build; and then integrates the build into the software structure. The build normally comprises a set of modules that perform a major function of the software system. Initially, the function may be represented by a stub that is replaced when the build is integrated.

In the top-down approach, individual stubs are replaced so that the top-level control is tested first, followed by stub replacements that move downward in the software structure. Using top-down integration, all modules that comprise a major function are integrated, thereby allowing an operational function to be demonstrated prior to completion of the entire system.

Work Products: At the completion of each level of integration testing, a test report is written. The report documents test results and lists any discrepancies that must be resolved before the tested components can be used as the foundation for another integration level. Place a copy of all integration test materials in the Project Test File.

A final test report is generated at the completion of integration testing indicating any unresolved difficulties that require management attention. Place a copy of the final Integration Test Report in the Project File.

Optional Work Product: A formal reporting system by which detected errors and discrepancies are recorded and fully described is recommended. These reports will help to confirm that all known errors are fixed before delivery of the completed software product. Error reports also help to trace multiple instances of the same error or anomalous behavior, so that error correction and prevention assignments can be implemented. The Quality Assurance representative assigned to the project can provide assistance in developing and using an error reporting/tracking system.

Activity: **8.2**
Conduct System Testing

Responsibility: Project Team or Independent Test Team

Description: During system testing, the completely integrated software product is tested to validate that the product meets all requirements. System response timing, memory, performance, security, and the functional accuracy of logic and numerical calculations are verified under both normal and high-load conditions. Query and report capabilities are exercised and validated. All operating documents are verified for completeness and accuracy.

System testing is conducted on the system testbed using the methodology and test cases described in the System Test Plan. The system testbed should be as close as possible to the actual production system. Either the project team or an independent test team conducts system testing to assure that the system performs as expected and that each function executes without error. The results of each test are recorded and upon completion included as part of the project test documentation.

When errors are discovered, they should be reviewed by the test team leader to determine the severity and necessary subsequent action. If appropriate, minor problems can be corrected and regression tested by the project team programmers within the time frame allotted for the system test. Any corrections or changes to the software product must be controlled under configuration management. Major problems may be cause to suspend or terminate the system test, which should then be rescheduled to begin after all of the problems are resolved.

Encourage users to participate in the system tests to gain their confidence in the software product and to receive an early indication of any problems from the user's perspective. Inform users that errors and discrepancies may occur during testing and explain the error correction, configuration management, and retest processes.

At the successful conclusion of system testing, the software product is ready for installation and acceptance testing.

Work Products: Review the draft versions of the operating documents, Training Plan, and Installation Plan. Update the documents as needed. Deliver the final versions of the operating documents, Training Plan, and Installation Plan to the system

***Work Products,
continued:***

owner and user for review and approval. Place a copy of the approved documents in the Project File. Place a copy of all system test materials (e.g., inputs, outputs, results, and error logs) in the Project Test File.

Generate a test report at the conclusion of the system test process. The report documents the system test results and lists any discrepancies that must be resolved before the software product is installed and prepared for acceptance testing. Place a copy of the report in the Project File.

Activity: **8.3**
Initiate Acceptance Process

Responsibility: Project Manager

Description: The acceptance process is used to officially accept new or modified software products that satisfy the project requirements and are fully operational. The initiation of the acceptance process begins after the successful completion of system testing. Prior to the initiation of the acceptance process, review the draft Acceptance Test Plan. Make any additions or changes needed to assure that the test plan reflects the current version of the software requirements.

The acceptance process is initiated with the completion of a Preacceptance Checklist. This list helps to ensure that all necessary preacceptance activities have been completed and that the required operating documents were developed and approved. The Preacceptance Checklist includes a section on software security issues.

Procedure: Use the following procedure to initiate the acceptance process.

- The project manager notifies the Quality Assurance Team assigned to the project that the project is ready to start the acceptance process.
- The Quality Assurance Team sends the Preacceptance Checklist to the project manager.
- The project manager completes the checklist, obtains the concurrence signature of the system owner (if required), and returns the completed checklist to the Quality Assurance Team.
- The Quality Assurance Team schedules an initial acceptance process meeting. More than one meeting may be necessary to accommodate users at different locations or with varying requirements.

Work Product: Review the draft version of the Acceptance Test Plan, and update as needed. Deliver the final version of the Acceptance Test Plan to the system owner, user, and other project stakeholders for review and approval prior to conducting any acceptance tests. Place a copy of the approved Acceptance Test Plan in the Project File. The Preacceptance Checklist is completed and submitted to the Quality Assurance Team supporting the project. A sample Preacceptance Checklist is provided on the following pages.

Sample Preacceptance Checklist
[Software Product Name]

Instructions: The project manager must indicate completion of each item with a checkmark and obtain concurrence on the last page. Send a copy of the completed checklist to the Quality Assurance Team assigned to the project. Any deviations from the checklist must be documented and a copy attached to the checklist.

1. Acceptance Test Plan

___ Provided to Quality Assurance and approved.

2. System Documents (e.g., User's Manual and Programmer's Reference Manual)

___ Appropriate system documents have been prepared in accordance with the applicable documentation standards. The system documents have been reviewed and approved by the system owner and other designated approvers. Issues identified during the Stage Exits have been resolved.

3. System Review Inventory System (SRIS)

___ The SRIS form has been updated to reflect the latest version of the software product and was submitted to the site software inventory/repository administrator and the Headquarters SRIS Coordinator.

4. Quality Assurance Support Preparation

___ The Quality Assurance Team has been given access to all portions of the software product required for testing.

___ Software product information has been provided to the Quality Assurance Team.

5. Security

___ The security checklist has been completed by the system owner and forwarded to the site's Computer Protection Program Manager. A sample security checklist is provided at the end of the Preacceptance Checklist.

___ Criteria for determination of mission essentiality have been reviewed and a determination made for the software product.

The software is mission essential. ☐ Yes ☐ No

___ If mission essential, a Continuity of Operations Plan is in place. The plan has been reviewed and approved by the appropriate organizations and personnel.

6. General Requirements

___ Installation of the required hardware and software for all users is fully documented in an Installation Plan.

___ User training is fully documented in an approved Training Plan.

___ Maintenance programmer staff training is fully documented in an approved Training Plan.

___ Configuration management is occurring according to the Software Configuration Management Plan.

___ Data retention criteria have been established according to appropriate recordkeeping requirements.

___ User identification has been assigned.

___ A task assignment has been generated to cover enhancement and maintenance services after acceptance.

Sample Preacceptance Checklist (continued)
[Software Product Name]

7. Software Requirements

- ___ All debugging and monitoring facilities have been removed from the production source and load (executable) modules.
- ___ Appropriate operational area points-of-contact have been consulted and agree that all operational readiness issues have been satisfied.

8. System Testing

System testing has been performed on all programs and modules to verify that the following conditions have been met.

- ___ User-required features have been satisfied (e.g., reports, data entry, data validation, queries).
- ___ All error conditions specified in the Integration and System Test Plans have been tested and respond to corrective action.
- ___ All backup, recovery, checkpoint, purge, and restart facilities required to ensure system integrity are operational.
- ___ Response times have been demonstrated and are in line with the requirements.
- ___ A production data base has been established.

System Owner Concurrence (if required)

I concur that all of the above items have been completed, and the system is ready for the acceptance process. Any deviations from the checklist have been documented and approved.

System Owner

Sample Preacceptance Checklist
Security Issues
[Software Product Name]

1. Sensitivity and Essentiality

The software is: ☐ non-sensitive ☐ sensitive
 ☐ classified ☐ mission essential

2. General Security (all software products)

Yes No

- | | | | |
|----|--------------------------|--------------------------|---|
| a. | <input type="checkbox"/> | <input type="checkbox"/> | Security objectives were established by the system owner. |
| b. | <input type="checkbox"/> | <input type="checkbox"/> | Security requirements were specified by the project team to support the security objectives. |
| c. | <input type="checkbox"/> | <input type="checkbox"/> | System design features enforce the security requirements. |
| d. | <input type="checkbox"/> | <input type="checkbox"/> | Testing was conducted to verify the security design features incorporated into the software product and the results were recorded. |
| e. | <input type="checkbox"/> | <input type="checkbox"/> | Security tests were conducted satisfactorily or a statement of acceptance of risk was issued by the system owner. |
| f. | <input type="checkbox"/> | <input type="checkbox"/> | Appropriate data set/file protection rules, authorities, and user identification codes were established by the system owner or as mandated by higher authority. |
| g. | <input type="checkbox"/> | <input type="checkbox"/> | Access control protection was incorporated into the software product. |
| h. | <input type="checkbox"/> | <input type="checkbox"/> | All manufacturer generic, test team, temporary, and superfluous passwords were deleted from the software product. |
| i. | <input type="checkbox"/> | <input type="checkbox"/> | All privacy, freedom of information, sensitivity, and classification considerations were identified, resolved, and established. |

3. Classified Software Products

Yes No

- | | | | |
|----|--------------------------|--------------------------|--|
| a. | <input type="checkbox"/> | <input type="checkbox"/> | An approved Security Plan was developed. |
| b. | <input type="checkbox"/> | <input type="checkbox"/> | For applications running on a classified system, the system owner has provided the required information to the Computer System Security Officer (CSSO) for the Computer Center for inclusion in the security plan. |
| c. | <input type="checkbox"/> | <input type="checkbox"/> | The security test plan was approved. |
| d. | <input type="checkbox"/> | <input type="checkbox"/> | The security test was successfully completed. |
| e. | <input type="checkbox"/> | <input type="checkbox"/> | The software product was certified by the owning organization's CSSO. |

Activity: **8.4**
Conduct Acceptance Test Team Training

Responsibility: Project Team

Description: If the project team is not conducting the Acceptance Test, training may be required for the personnel performing the testing. The acceptance test participants and their experience with the software product and the operating environment should have been identified in the Acceptance Test Plan.

The level of training will depend on the testers' familiarity with the software product and the platform on which the software will run. The advantage of having users acceptance test the software product is that they are the experts most familiar with the business information flow and how the software product must fit into the workplace.

It is recommended that the operating documents and other test materials be distributed to the test team prior to the actual start of the acceptance test training. This will give the test team time to become familiar with the software product and the test process and procedures.

Activity: **8.5**
Revise Project Plan

Responsibility: Project Manager

Description: Once the integration and system tests are completed, determine if the project estimates for resources, cost, and schedule need to be revised.

Work Product: Review the Project Plan for accuracy and completeness of all Software Integration and Testing Stage activities and make any changes needed to update the information. Expand the information for the Installation and Acceptance Stage to reflect accurate estimates of resources, costs, and hours. Place a copy of the revised Project Plan in the Project File.

Note: A Project Plan is an effective management tool that is recommended for all projects regardless of size. The plan can be consolidated for small projects.

Review Process: Conduct a structured walkthrough to ensure that the Project Plan reflects the project's current status and adequately estimates the resources, costs, and schedule for the Installation and Acceptance Stage.

The Project Plan is formally reviewed during the In-Stage Assessment and Stage Exit processes.

Activity: **8.6
Conduct In-Stage Assessment**

Responsibility: Project Manager and Independent Reviewer

Description: An In-Stage Assessment (ISA) is an independent review of the work products and deliverables developed or revised during each stage of the project lifecycle. The independent reviewer is typically a member of the Quality Assurance Team who is assigned to the software project and conducts all of the ISAs for the project.

An ISA does not require meetings with, or extra work by, the project team. All of the work products and deliverables needed for the review should be readily available in the Project File.

Schedule at least one ISA prior to the Software Integration and Testing Stage Exit process. Additional ISAs can be performed during the stage, as needed. Periodic reviews of the integration and system test results and logs are recommended.

Provide the reviewer with copies of all work products developed or revised during the Software Integration and Testing Stage including the Project Plan. The reviewer assesses the work products and deliverables to verify the following:

- The project is complying with the site's software engineering standards/best practices.
- Sound project management practices are being used.
- Project risks are identified and mitigated.

A description of the ISA process and the ISA report form are provided in the *In-Stage Assessment Process Guide*. A copy of the guide is provided in Appendix D.

Note: An ISA is an effective project management tool that is recommended for all projects regardless of size.

Work Product: An ISA report form is prepared by the independent reviewer and is used to identify open issues that need to be resolved in this stage. The report is delivered to the project manager and a copy should be placed in the Project File.

Activity:	8.7 Conduct Software Integration and Testing Stage Exit
Responsibility:	Project Manager
Description:	<p>The Stage Exit is a process for ensuring that projects are on target, within budget, on schedule, and meet the DOE and project standards identified in the Project Plan. The goal of a Stage Exit is to secure the approval of designated key individuals to continue with the project and to move forward into the next lifecycle stage.</p> <p>Schedule the Stage Exit as the last activity of the Software Integration and Testing Stage. It is the responsibility of the project manager to notify the appropriate participants when a project is ready for the Stage Exit process and to schedule the Stage Exit meeting. All functional areas and the Quality Assurance representative involved with the project should receive copies of the work products and deliverables produced in this stage.</p> <p>During the Stage Exit meeting, participants discuss open issues that will impact the Project Plan. The project manager should ensure that an acceptable action plan is developed for handling all open issues. At the conclusion of the meeting, concurrence is needed from the designated approvers to begin the next stage.</p> <p>A description of the Stage Exit process is provided in the <i>Stage Exit Process Guide</i>. A copy of the guide is provided in Appendix E.</p>
Note:	A Stage Exit is an effective project management tool that is recommended for all software projects regardless of size. For small software projects, stages can be combined and addressed during one Stage Exit.
Work Product:	A summary of the Stage Exit meeting is prepared by the project manager or a designee and distributed to the meeting attendees. The summary identifies any issues and action items needed to obtain concurrence prior to proceeding to the Installation and Acceptance.